МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный университет имени М.В.Ломоносова»

«Утверждаю»

Декан факультета ВМК МГУ
имени М.В. Ломоносова

академик _____ Е. И. Моисеев

«___» _____ 2017 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Конструирование компиляторов»

Уровень высшего образования – подготовка научно-педагогических кадров в аспирантуре

Направление подготовки–09.06.01 «Информатика и вычислительная техника»

Направленность (профиль)–«Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» (05.13.11)

2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

1. НАИМЕНОВАНИЕ ДИСЦИПЛИНЫ

Конструирование компиляторов

2. УРОВЕНЬ ВЫСШЕГО ОБРАЗОВАНИЯ

Подготовка научно-педагогических кадров в аспирантуре.

3. НАПРАВЛЕНИЕ ПОДГОТОВКИ, НАПРАВЛЕННОСТЬ (ПРОФИЛЬ) ПОДГОТОВКИ

Направление 09.06.01 «Информатика и вычислительная техника». Направленность (профиль) «Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей» (05.13.11).

4. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина относитсякспециальным дисциплинам вариативной части образовательной программы и относится к вариативной части.

5.ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ

Дисциплинаучаствуетв формировании следующих компетенцийобразовательной программы:

Формируемые компетенции	Планируемые результаты обучения
Владениемметодологией теоретических и экспериментальныхисследований в областипрофессиональной деятельности (ОПК-1)	З1 (ОПК-1) ЗНАТЬ: современные математические методы, применяющиеся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий У1 (ОПК-1) УМЕТЬ: применять современные методы постановки и анализа задач в области математики и информатики В1 (ОПК-1) ВЛАДЕТЬ: навыками оптимального выбора современных методов и средств постановки и анализа задач в области математики и информатики

Способность разрабатывать и реализовывать алгоритмы организации работы современных вычислительных комплексов и компьютерных сетей (ПК-2)	З1 (ПК-2) ЗНАТЬ: современные методыразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения У1(ПК-2) УМЕТЬ: применять современные методыразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения В1 (ПК-2) ВЛАДЕТЬ: навыками оптимального выбора современных методовразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения
Способность к реализации различных математических алгоритмов в виде программных комплексов, ориентированных на современную вычислительную технику (ПК-4)	З1 (ПК-4)ЗНАТЬ: современные методы реализации различных математических алгоритмов в виде программных комплексов, особенности современных вычислительных комплексов У1(ПК-4) УМЕТЬ: применять современные методы реализации различных математических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов В1 (ПК-4) ВЛАДЕТЬ: навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов

Оценочные средства для промежуточной аттестации приведены в Приложении.

6. ОБЪЕМ ДИСЦИПЛИНЫ

Объем дисциплины составляет 3 зачетных единицы, всего 108 часов.

38 часов составляет контактная работа с преподавателем – 32 часа занятий лекционного типа, 0 часов занятий семинарского типа (семинары, научно-практические занятия, лабораторные работы и т.п.), 0 часов индивидуальных консультаций, 2 часа мероприятий текущего контроля успеваемости, 2 часа групповых консультаций, 2 часа мероприятий промежуточной аттестации.

70 часов составляет самостоятельная работа аспиранта.

7. ВХОДНЫЕ ТРЕБОВАНИЯ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Учащиеся должны владеть знаниями поалгоритмам, алгоритмическим языкам и программированию в объеме, соответствующемосновным образовательным программам бакалавриата и магистратуры по укрупненным группам направлений и специальностей 01.00.00 «Математика и механика», 02.00.00 «Компьютерные и информационные науки».

8. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При изложении материала лекций предполагается диалог со слушателями. На каждой лекции выделяется время для ответов на вопросы по текущему материалу и его обсуждения. Материалы лекций демонстрируются аспирантам в виде презентаций, сопровождаемых комментариями лектора. Дополнительно каждый аспирант может дистанционно получить разъяснения преподавателя по электронной почте.

9. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Освоение дисциплины «Конструирование компиляторов» обеспечивает обучающихся необходимыми знаниями и навыками в области оптимизирующей компиляции, а также в таких смежных областях, как статический анализ для выявления дефектов, обратная инженерия, генерация тестовых покрытий и др.. В рамках курса предусмотрено обучение и закрепление навыков применения изученного ранее математического аппарата для решения прикладных задач.

Целью освоения дисциплины является получение базовых знаний в области разработки и применения современных компиляторов и компиляторных сред для разработки программ и для решения некоторых задач по обеспечению безопасного функционирования программ, для решения которых применяются компиляторные технологии.

Наименование и краткое содержание разделов и тем дисциплины (модуля), форма промежуточной аттестации по дисциплине (модулю)	**Всего (часы)**	В том числе								
		Контактная работа (работа во взаимодействии с преподавателем), часы из них						**Самостоятельная работа обучающегося, часы** из них		
		Занятия лекционного типа	Занятия семинарского типа	Групповые консультации	Индивидуальные консультации	Учебные занятия, направленные на проведение текущего контроля успеваемости (коллоквиумы, практические контрольные занятия и др)*	**Всего**	Выполнение домашних заданий	Подготовка рефратов и т. п..	**Всего**
Тема 1.*Введение*(Описание процесса компиляции. Структура оптимизирующего компилятора. Основные вопросы, изучаемые в курсе.)	4	2	0	0	0	0	2	2	0	2
Тема 2. *Построение промежуточного представления программы*(Базовые блоки и граф потока управления. Биткод среды LLVM – пример промежуточного представления.)	4	2	0	0	0	0	2	2	0	2
Тема 3. *Локальная оптимизация*(Метод нумерации значений. Представление базового блока в виде направленного ациклического графа.)	4	2	0	0	0	0	2	2	0	2

Тема 4.*Анализ потока данных*(Основной метод глобальной оптимизации. Примеры анализа потока данных – анализ достигающих определений и анализ живых переменных.)	4	2	0	0	0	0	2	2	0	2
Тема 5.*Граф потока управления*(Остовное дерево, его обход, нумерация вершин, классификация дуг, отношение доминирования и построение дерева доминаторов.)	5	3	0	0	0	0	3	2	0	2
Тема 6.*SSA-форма промежуточного представления*(ПостроениеSSA-формы промежуточного представления. Граница доминирования. Анализ потока данных в SSA-форме. Доступные выражения.)	5	3	0	0	0	0	3	2	0	2
Тема 7.*Обоснование анализа потока данных*(Полурешетки, передаточные функции, общий итерационный алгоритм.)	5	3	0	0	0	0	3	2	0	2
Тема 8.*Методы ускорения анализа потока данных.*(Суперблоки и другие области графа потока управления. Вычисление передаточных функций областей по передаточным функциям составляющих их базовых блоков.)	5	3	0	0	0	0	3	2	0	2
Тема 9.*Глобальный метод нумерации значений* (Содержание шагов ме-	4	2	0	0	0	0	2	2	0	2

тода.)										
Тема 10.*Глобальный анализ указателей*(Псевдонимы / алиасы. Недостаточность глобального анализа. Межпроцедурный анализ. Граф вызовов. Методы учета контекста.)	5	3	0	0	0	0	3	2	0	2
Тема 11.*Задачи, решаемые на этапе машинно-ориентированной оптимизации*(Планирование кода. Распределение регистров.)	5	3	0	0	0	0	3	2	0	2
Тема 12.*Другие методы оптимизации*(Оптимизация потока управления, возвраты из рекурсивных функций. Раскрутка циклов. Открытая вставка функций.)	4	2	0	0	0	0	2	2	0	2
Тема 13.*Генерация объектного кода методом переписывания дерева*(Содержание шагов метода переписывания дерева при генерации объектного кода)	4	2	0	0	0	0	2	2	0	2
14. Текущий контроль успеваемости: ***защита практического задания***	10	0	0	2	0	0	2	8		
15. Промежуточная аттестация: ***устный экзамен***	40	0	0	2	0	2	4	36		
Итого	108	32	0	4	0	2	38	70		

10. УЧЕБНО-МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ УЧАЩИХСЯ

Самостоятельная работа учащихся состоит в изучении лекционного материала, учебно-методической литературы, выполнения домашних заданий, выполнения практического задания, составления отчёта по практическому заданию, подготовки к текущему контролю и к промежуточной аттестации:

• подготовкик защите практического задания;

• подготовки к устному экзамену.

Указания по выполнению практического задания

В качестве практического задания студентам предлагается попробовать самостоятельно реализовать одно из оптимизирующих преобразований, используя инфраструктуру компилятора LLVM и его промежуточное представление – биткод.

Примеры оптимизирующих преобразований, предлагаемых для реализации в рамках практического задания:

- нахождение доступных выражений (для исключения избыточных вычислений);
- планирование кода в суперблоке;
- распространение копий;
- вынос инвариантных вычислений в предзаголовок цикла;
- распространение констант (с вычислением или без вычисления);

Приступая к выполнению задания, аспирант должен изучить среду LLVM, ее промежуточное представление (биткод), и, воспользовавшись компилятором Clang, получить биткод своего задания. Далее, пользуясь возможностями LLVM, он должен составить фазу (программу), выполняющую требуемое преобразование и включить ее в состав учебного компилятора на базе LLVM. Для проверки задания используется автоматическая система на удаленном сервере. В завершении задания составляется итоговый отчёт. Проводится защита выполненного задания перед комиссией преподавателей.

Срок выполнения практического задания – 12 недель. В конце семестра по итогам защиты выполненного задания выставляются итоговые технические баллы.

Указания по составлению отчёта по практическому заданию

Отчёт пишется на русском языке. Вёрстку можно осуществлять в любой подходящей для Вас системе. Текст отчёта должен быть разбит на следующие части:

• Титульный лист, с «шапкой» – «Московский государственный университет имени М. В. Ломоносова, факультет Вычислительной математики и кибернетики». Далее следует заголовок: «Отчёт по практическому заданию», номер и тема варианта задания, сведения об исполнителе (фамилия, имя и отчество полностью, номер группы). Внизу титульного листа указывается город и год. Нелишне обратить внимание на то, что точки после заголовков не ставятся.
• Содержание, которое состоит из перечня названий глав и подглав, сопровождаемых указанием номеров страниц, с которых они начинаются. Нумеруются все страницы, за исключением титульного листа. Номер страницы с содержанием: 2.
• Первая глава, названная «Постановка задачи», содержит формулировку задания и описание оптимизирующего преобразования, которое предстоит реализовать в рамках него. Каждую главу следует начинать с новой страницы.
• Вторая глава, названная «Теоретические аспекты», содержит описание идей, методов, алгоритмов, подходов, привлекаемых при выполнении задания.
• Третья глава, названная «Практические аспекты», содержит описание реализованного программного кода. В главе обязательно должна быть приведена иллюстрация с модульной / компонентной структурой реализованного кода. По каждому модулю / компоненту кода должны быть даны пояснения о том, каково его назначение, каковы его связи с другими модулями / компонентами, каковы его внутренние составляющие части как видимые извне, так и скрытые.
• Четвёртая глава, названная «Результаты», содержит анализ результатов работы реализованной программы. Следует охарактеризовать результаты, полученные на тестовых прогонах программы, оценить их. Если доступны другие реализации указанного в заданииоптимизирующего преобразования, следует сравнить результаты их работы с результатами, демонстрируемыми Вашей программой.
• Заключение (которое не нумеруется, но номер на странице ставится), где подводится общий итог работы, завершает отчёт. В заключении можно указать характеристики написанного кода, привести соображения о том, насколько удачно удалось справиться срешеним доставшейся Вам задачи.
• Список использованной литературы приводится, если в ходе работы над заданием были использованы статьи и/или книги. Библиографические записи в списке следует оформлять по рекомендациям ГОСТ. Сделать это можно при помощи Google.Schollar, который умеет импортировать по ГОСТ. На каждую запись списка в тексте отчёта должна быть ссылка.
• Приложение, которое содержит Ваш код.

11.РЕСУРСНОЕ ОБЕСПЕЧЕНИЕ

Основная литература

1. KeithD. Cooper, LindaTorczon. Engineering a Compiler, Second Edition. 2012 Elsevier, Inc..
2. Сети Р., Лам М. С., Ульман Дж. Д., Ахо А. В. Компиляторы. Принципы, технологии и инструментарий – М.: «Вильямс» – 2016

Дополнительная литература

1. Steven S. Muchnick Advanced Compiler Design & Implementation. MorganKaufmanPublishers, 1997
2. Y.N. Srikant, Priti Shankar. The Compiler Design Handbook, Second Edition, CRC Press, 2008.

Ресурсы информационно-телекоммуникационной сети «Интернет»

1. Веб-страница дисциплины: http://algcourse.cs.msu.su/?page_id=409
2. Веб-сайт LLVM http://llvm.org

Информационные технологии, используемые в процессе обучения

1. Программное обеспечение для подготовки слайдов лекцийOpenOfficeImpress.
2. Программное обеспечение для создания и просмотра pdf-документов AdobeReader.
3. ПлатформаLLVM
4. Веб-браузер для доступа к материалам, размещённым в WWW (MozillaFirefoxили аналогичный).
5. Программа-клиент электронной почты для онлайновых консультаций с лектором (MozillaThunderbirdили аналогичный).

Активные и интерактивные формы проведения занятия

№ п\п	Тип занятия или внеаудиторной работы	Вид и тематика (название) интерактивного занятия
1	Лекции по темам №№ 1-13.	В рамках каждого лекционного занятия выделяется время (около 15 минут) для интерактивного обсуждения материала.
2	Текущий контроль	Текущий контроль осуществляется в виде защиты выполненного практического задания пе-

		ред комиссией. Аспирант готовит мультимедийную презентацию, составляет текстовый отчёт и делает устное сообщение о проделанной им работе по заданию. После устного сообщения комиссия задаёт вопросы и предоставляет возможность ответить на них.

Материально-техническая база

Для преподавания дисциплины требуется класс, оборудованный маркерной или меловой доской и мультимедийным проектором. Для выполнения домашних заданий, а также для выполнения и сдачи практических заданий необходим персональный компьютер с доступом в Internet и установленной на нём компиляторной платформой LLVM.

12. ЯЗЫК ПРЕПОДАВАНИЯ

Русский

13. РАЗРАБОТЧИК ПРОГРАММЫ, ПРЕПОДАВАТЕЛИ

профессор кафедры, к.ф.-м.н.Гайсарян Сергей Суренович

Приложение

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

«Конструирование компиляторов»

Средства для оценивания планируемых результатов обучения, критерии и показатели оценивания приведены ниже.

РЕЗУЛЬТАТ ОБУЧЕНИЯ по дисциплине (модулю)	**КРИТЕРИИ и ПОКАЗАТЕЛИ ОЦЕНИВАНИЯ РЕЗУЛЬТАТА ОБУЧЕНИЯ по дисциплине (модулю)** *(критерии и показатели берутся из соответствующих карт компетенций, при этом пользуются либо традиционной системой оценивания, либо БРС)*					**ОЦЕНОЧНЫЕ СРЕДСТВА**
	1	2	3	4	5	
	Неудовлетворительно	**Неудовлетворительно**	**Удовлетворительно**	**Хорошо**	**Отлично**	
ЗНАТЬ: современные математические методы, применяющиеся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий **Код З1 (ОПК-1)**	Отсутствие знаний	Фрагментарные представления о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	В целом сформированные, но неполные знания о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	Сформированные, но содержащие отдельные пробелы знания о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	Сформированные систематические знания о современных математических методах, применяющихся для решения задач в области естественных наук, экономики, социологии и информационно-коммуникационных технологий	Устный экзамен
УМЕТЬ: применять современные методы постановкии анализа задач в	Отсутствие умений	Фрагментарные умения применять современные методы постановкии анализа задач в	В целом успешное, но не систематическое умение применять современные	Успешное, но содержащее отдельные пробелы умение применять со-	Сформированное умение применять современные методы постановкии анализа задач в	Устный экзамен

области математики и информатики **Код У1 (ОПК-1)**		области математики и информатики	методы постановкии анализа задач в области математики и информатики	временные методы постановкии анализа задач в области математики и информатики	области математики и информатики	
ВЛАДЕТЬ: навыками оптимального выбора современных методов и средств постановкии анализа задач в области математики и информатики **Код В1 (ОПК-1)**	Отсутствие навыков	Фрагментарноевладение навыками оптимального выбора современных методов и средств постановкии анализа задач в области математики и информатики	В целом успешное, но не полное владение навыками оптимального выбора современных методов и средств постановкии анализа задач в области математики и информатики	Успешное, но содержащее отдельные пробелывладение навыками оптимального выбора современных методов и средств постановкии анализа задач в области математики и информатики	Сформированное владение навыками оптимального выбора современных методов и средств постановкии анализа задач в области математики и информатики	Устный экзамен
ЗНАТЬ: современные методы разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения **Код З1 (ПК-2)**	Отсутствие знаний	Фрагментарные представления о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	В целом сформированные, но неполные знания о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированные, но содержащие отдельные пробелы знания о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированные систематические знания о современных методах разработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Устный экзамен
УМЕТЬ: применять современные методыразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Отсутствие умений	Фрагментарные умения применять современные методыразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	В целом успешное, но не систематическое умение применять современные методыразработки и реализации алгоритмов организации работы вычислительных ком-	Успешное, но содержащее отдельные пробелы умение применять современные методыразработки и реализации алгоритмов организации работы вычислительных	Сформированное умение применять современные методыразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Отчет

Код У1 (ПК-2)			плексов и компьютерных сетей последнего поколения	комплексов и компьютерных сетей последнего поколения		
ВЛАДЕТЬ: навыками оптимального выбора современных методовразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения **Код В1 (ПК-2)**	Отсутствие навыков	Фрагментарноевладение навыками оптимального выбора современных методовразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	В целом успешное, но не полное владение навыками оптимального выбора современных методовразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Успешное, но содержащее отдельные пробелывладение навыками оптимального выбора современных методовразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	Сформированное владение навыками оптимального выбора современных методовразработки и реализации алгоритмов организации работы вычислительных комплексов и компьютерных сетей последнего поколения	отчет
ЗНАТЬ: современные методы реализации различных математических алгоритмов в виде программных комплексов, особенности современных вычислительных комплексов **Код З1 (ПК-4)**	Отсутствие знаний	Фрагментарные представления осовременных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	В целом сформированные, но неполные знания осовременных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	Сформированные, но содержащие отдельные пробелы знания осовременных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	Сформированные систематические знания осовременных методах реализации различных математических алгоритмов в виде программных комплексов, особенностях современных вычислительных комплексов	Устный экзамен
УМЕТЬ: применять современные методы реализации различных математических алгоритмов в виде программ-	Отсутствие умений	Фрагментарные умениеприменять современныеметоды реализации различных математических алгоритмов в виде программ-	В целом успешное, но не систематическое умениеприменять современныеметоды реализации различных матема-	Успешное, но содержащее отдельные пробелы умениеприменять современныеметоды реализации различ-	Сформированное умение применять современныеметоды реализации различных математических алгоритмов в виде программ-	отчет

ных комплексов с учетом особенностей современных вычислительных комплексов **Код У1 (ПК-4)**		ных комплексов с учетом особенностей современных вычислительных комплексов	тических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов	ных математических алгоритмов в виде программных комплексов с учетом особенностей современных вычислительных комплексов	ных комплексов с учетом особенностей современных вычислительных комплексов	
ВЛАДЕТЬ: навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов **Код В1 (ПК-4)**	Отсутствие навыков	Фрагментарноевладение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	В целом успешное, но не полное владение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	Успешное, но содержащее отдельные пробелывладение навыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	Сформированное владениенавыками оптимального выбора и создания новых современных методов реализации математических алгоритмов в виде программных комплексов, учитывающих особенности современных вычислительных комплексов	отчет

Фонды оценочных средств, необходимые для оценки результатов обучения

Типовые контрольные задания или иные материалы для проведения текущего контроля успеваемости.
Пример постановки задачи для практического задания по реализации оптимизирующего преобразования

Постановка задачи

Целью работы является создание оптимизирующего преобразования, позволяющего повысить качество кода, генерируемого учебным компилятором на базе LLVM. Задача оптимизации кода состоит в повышении его быстродействия и/или сокращения его размера.

Предлагается разработать преобразование, которое на вход получает программу в промежуточном представлении LLVM, а на выходе генерирует ее версию, оптимизированную по размеру и скорости выполнения путем удаления недостижимого и бесполезного кода. Например:

Первоначальная программа:

```
intmain (void) {
printf(“Firstmessage \n”);
int a = 34+48;
if (0)
printf(“Unreachable code\n”);
a = a * 2;
printf(“Second message\n”);
return 0;
}
```

Программа после удаления мертвого и недостижимого кода:

```
int main (void) {
printf(“First message \n”);
printf(“Second message\n”);
return 0;
}
```

Решение задачи

Теоретические аспекты

Предполагается реализация алгоритмов удаления мертвого и недостижимого кода. Недостижимым кодом называют часть кода программы, которая ни при каких условиях не может быть исполнена, поскольку является недостижимой в графе потока управления. Мертвый (неиспользуемый, бесполезный) код - команды, вычисляющие никогда не использующиеся значения.

Практические аспекты

Решения должны быть написаны на языке C++ с использованием контейнеров и алгоритмов стандартной библиотеки, а также средств, предоставляемых LLVM. Оптимизирующие преобразования должны быть выполнены в учебном компиляторе, который представляет собой модифицированную версию LLVM, не содержащую оптимизирующих преобразований.

Пример сборки LLVM в ОС Linux:

В каталоге с исходными кодами следует создать каталог с названием build и перейти в него:

```
$ cd llvm/
$ mkdir build
$ cd build
```
Затем следует осуществить сборку LLVM с указанием каталога сборки и желаемого типа сборки (Debug/Release). Сборка осуществляется с помощью утилит configure и make. Ключи configure:
--prefix=<путь к каталогу инсталляции>
--disable-assertions/--enable-assertion – включить/выключить проверку утверждений
--enable-optimized/--disable-optimized – выбрать тип сборки: оптимизированная (Release) или без оптимизаций (Debug)

Для ускорения компиляции рекомендуется использовать ключ “-jN” утилиты make, где N -- число желаемых потоков сборки (как правило, равное количеству ядер процессора или количеству ядер процессора + 1).
Пример сборки Debug с выключенной проверкой утверждений и установкой в каталог </home/user/llvm> и сборкой в 8 потоков:
$ ../configure –prefix=/home/user/llvm –disable-optimized – disable-assertions
$ make –j8
$ makeinstall
После завершения установки и компиляции требуется добавить путь до каталога с LLVM в переменную окружения PATH:
$ export PATH=/home/user/llvm/bin:$PATH
Получить промежуточное представление LLVM для программы можно, выполнив команду:
$ clang –c –O0 –emit-llvmtest.c –o test.bc
Перевод бинарного представления в ассемблер LLVM
$ llvm-dis test.bc –o test.ll
Ассемблирование в бинарное представление
$ llvm-astest.ll –onewTest.bc
Запуск преобразования pass_name из динамической библиотеки pass_name.so
$ opt -load<путь/до/динамической/библиотеки/>pass_name.so -<pass_name>test.bc –otransformedTest.bc
test.bc - файл, содержащий бинарную версию промежуточного представления LLVM.
test.ll – файл, содержащий ассемблер LLVM в читаемом виде.
Написание оптимизирующего прохода LLVM
Для выполнения работы могут потребоваться проходы следующих типов: FunctionPass (по функциям), ModulePass (по модулям), BasicBlockPass (по базовым блокам).

В качестве примера приведен проход по функциям и именем Hello, вызываемым из командной строки с помощью ключа “–hello”:

```
#include "llvm/Pass.h"
#include "llvm/IR/Function.h"
#include "llvm/Support/raw_ostream.h"
using namespace llvm;
namespace {
struct Hello : public FunctionPass {
static char ID;
Hello() : FunctionPass(ID) {}
virtual bool runOnFunction(Function &F) {
errs() << "Hello: ";
errs().write_escaped(F.getName()) << '\n';
return false;
 }
 };
}
char Hello::ID = 0;
staticRegisterPass<Hello> X("hello", "Hello World Pass", false, false);
```

Листинг 1. Пример компиляторного прохода

Данный код выводит на экран сообщения вида “Hello: <function_name>”.

Более детально изучить построение оптимизирующих проходов можно посмотреть в документации на компилятор (страница WritingAnLLVMPass.html в каталоге docs в поставке учебного компилятора).

Тестирование

На личной странице расположена форма загрузки файла, а также информация о результатах тестирования и минимальный набор синтетических тестов.

Загрузка решения. Загружаемый файл должен представлять собой текст программы на языке C++, содержащий компиляторный проход, запускаемый по ключу “-dce”. После загрузки решение будет скомпилировано и запущенно с помощью утилиты “opt” на тестовом наборе данных.

Оценивается последнее присланное решение.

Для тестирования на локальной машине во время разработки предлагается осуществлять компиляцию программы с уровнем оптимизации “O0”. Локальное тестирование предлагается осуществлять с помощью программ с открытыми исходными кодами и минимального набора синтетических тестов.

Оценка

Обязательное условие:

Присланное решение должно проходить проверку на корректность на программах SQlite и Lzma.

Оценка эффективности будет производиться на расширенном синтетическом наборе тестов.

Необходимым условием получения оценки «Удовлетворительно» является реализация удаления бесполезного кода на уровне функций.

Необходимым условием получения оценки «Хорошо» является реализация удаления бесполезного и недостижимого кода на уровне функций. Удаление избыточных вызовов “pure” функций (не имеющих побочных эффектов и зависящих только от аргумента), например в выражении doubley=sin(x), если у нигде более не используется вызов sin(x) можно удалить.

Необходимым условием получения оценки «Отлично» является реализация удаления бесполезного и недостижимого кода на уровне функций. Реализован алгоритм удаления частично избыточного кода, алгоритм описан в публикации: JensKnoop, OliverRüthing, andBernhardSteffen. 1994. Partial dead code elimination.SIGPLAN Not. 29, 6 (June 1994), 147-158. DOI=10.1145/773473.178256 http://doi.acm.org/10.1145/773473.17825.

Итоговая оценка определяется по результатам защиты выполненного задания перед комиссией. На защиту предоставляется письменный отчёт, мультимедийная презентация, сопровождаемая устными пояснениями.

Типовые контрольные задания или иные материалы для проведения промежуточного контроля успеваемости.

Программа устного экзамена по дисциплине

1. Структура оптимизирующего компилятора. Построение промежуточного представления программы.
2. Базовые блоки и граф потока управления. Биткод среды LLVM – пример промежуточного представления.
3. Локальная оптимизация. Метод нумерации значений: представление базового блока в виде направленного ациклического графа.
4. Анализ потока данных – основной метод глобальной оптимизации. Примеры анализа потока данных – анализ достигающих определений
5. Анализ живых переменных. Исключение мерьвого кода.
6. Вынесение инвариантных вычислений за пределы цикла.
7. Граф потока управления: остовное дерево, обход, нумерация вершин, классификация дуг.
8. Отношение доминирования и построение дерева доминаторов
9. Построение естественных циклов и гнезд циклов.
10. SSA-форма промежуточного представления и ее построение. Граница доминирования.
11. Анализ потока данных в SSA-форме. Выявление доступных выражений. Исключение избыточности.
12. Обоснование анализа потока данных: полурешетки, передаточные функции, общий итерационный алгоритм.

13. Методы ускорения анализа потока данных. Суперблоки и другие области графа потока управления.
14. Вычисление передаточных функций областей по передаточным функциям составляющих их базовых блоков. Пример – анализ достигающих определений.
15. Вычисление передаточных функций областей по передаточным функциям составляющих их базовых блоков на примере анализа достигающих определений.
16. Глобальный метод нумерации значений – использование дерева доминаторов.
17. Глобальный анализ указателей. Псевдонимы (алиасы). Недостаточность глобального анализа.
18. Межпроцедурный анализ. Использование графа вызовов.
19. Межпроцедурный анализ. Методы учета контекста вызова.
20. Задачи, решаемые на этапе машинно-ориентированной оптимизации.
21. Планирование кода.
22. Распределение регистров.
23. Оптимизация потока управления, возвраты из рекурсивных функций.
24. Раскрутка циклов.
25. Открытая вставка функций.
26. Генерация объектного кода методом переписывания дерева

Методические материалы для проведения процедур оценивания результатов обучения

Система контроля и оценивания

Оценка по курсу устанавливается в зависимости от суммы технических баллов, набранных слушателем в ходе семестра. По итогам выполнения и защиты практического задания можно заработать до 60 технических баллов, за устный экзамен – до 40 технических баллов. Технические баллы за практическое задание выставляются в зависимости от качества решения задачи (метрика качества зависит от задания), соблюдения графика сдачи этапов задания в проверяющую систему, скрупулёзности составления и оформления отчёта по заданию, подробности и доходчивости представления выполненной работы на защите. Таким образом, максимально возможная сумма набранных технических баллов составляет 100. Оценка «отлично» ставится слушателям, набравшим от 80 баллов и выше. Оценка «хорошо» ставится слушателям, набравшим от 60 до 79 технических баллов. Оценка «удовлетворительно» ставится слушателям, набравшим от 40 до 59 технических баллов. Оценка «неудовлетворительно» ставится слушателям, набравшим менее 40 технических баллов.

Структура и график контрольных мероприятий

Сдача этапов практического задания в автоматическую проверяющую систему, представление письменного отчёта по практическому заданию и защита выполненного задания в конце семестра, устный экзамен в конце семестра.